

**IN THE CLAIMS:**

1. (Previously Presented): A method in a computer system executing a Web-based application, said method comprising the steps of:

associating one of a plurality of different priorities with each one of a plurality of different HTTP requests that are processed by an application;

establishing a plurality of different, separate queues;

associating each one of said plurality of different queues with a different one of said plurality of priorities;

for each one of said plurality of HTTP requests, storing one of said plurality of HTTP requests in one of said plurality of different queues that is associated with one of said plurality of priorities that is associated with said one of said plurality of HTTP requests, wherein all of said plurality of HTTP requests that are associated with a first one of said plurality of priorities are stored in a first one of said plurality of different queues that is associated with said first one of said plurality of priorities, and all of said plurality of HTTP requests that are associated with a second one of said plurality of priorities are stored in a second one of said plurality of different queues that is associated with said second one of said plurality of priorities;

completing processing of multiple ones of said plurality of HTTP requests that are stored in said first one of said plurality of different queues before beginning processing of multiple ones of said plurality of HTTP requests that are stored in said second one of said plurality of different queues regardless of whether or not said first one of said plurality of priorities is a higher priority than said second one of said plurality of priorities.

2. (Previously Presented): The method according to claim 1, further comprising the steps of:

associating one of a plurality of types of requests with each one of said plurality of different priorities;

identifying a type of each one of said plurality of HTTP requests; and

for each one of said plurality of HTTP requests, determining one of said plurality of different priorities associated with a type that was determined for each of said plurality of HTTP requests.

3. (Canceled)

4. (Canceled)

5. (Previously Presented): The method according to claim 1, further comprising the steps of:

receiving said plurality of HTTP requests by said application; and

determining one of said plurality of different priorities that is associated with a type of each one of said plurality of HTTP requests.

6. (Previously Presented): The method according to claim 1, further comprising the steps of:

receiving one of said plurality of HTTP requests by said application;

determining whether there is a backlog of pending HTTP requests waiting to be processed by said application;

in response to a determination that there is no backlog, immediately processing said one of said plurality of HTTP requests;

in response to a determination that there is a backlog, determining a type of said one of said plurality of HTTP requests;

identifying one of said plurality of priorities that is associated with said type;

identifying one of said plurality of queues that is associated with said identified one of said plurality of priorities; and

storing said one of said plurality of HTTP requests in said identified one of said plurality of queues.

7. (Canceled)

8. (Previously Presented): The method according to claim 1, further comprising the steps of:

storing multiple ones of said plurality of requests having a type associated with a high priority in one of said plurality of queues that is associated with said high priority;

storing multiple ones of said plurality of requests having a type associated with a low priority in one of said plurality of queues that is associated with said low priority; and

completing processing of said multiple ones of said plurality of requests stored in said one of said plurality of queues that is associated with said high priority before beginning processing of said multiple ones of said plurality of requests stored in said one of said plurality of queues that is associated with a low priority.

9. (Previously Presented): A computer program product encoded on a computer readable medium in a computer system executing a Web-based application, comprising:

computer executable instruction means for associating one of a plurality of different priorities with each one of a plurality of different HTTP requests that are processed by an application;

computer executable instruction means for establishing a plurality of different, separate queues;

computer executable instruction means for associating each one of said plurality of different queues with a different one of said plurality of priorities;

for each one of said plurality of HTTP requests, computer executable instruction means for storing one of said plurality of HTTP requests in one of said plurality of different queues that is associated with one of said plurality of priorities that is associated with said one of said plurality of HTTP requests, wherein all of said plurality of HTTP requests that are associated with a first one of said plurality of priorities are stored in a first one of said plurality of different queues that is associated with said first one of said plurality of priorities, and all of said plurality of HTTP requests that are associated with a second one of said plurality of priorities are stored in a second one of said plurality of different queues that is associated with said second one of said plurality of priorities;

computer executable instruction means for completing processing of multiple ones of said plurality of HTTP requests that are stored in said first one of said plurality of different queues before beginning processing of multiple ones of said plurality of HTTP requests that are stored in said second one of said plurality of different queues regardless of whether or not said first one of said plurality of priorities is a higher priority than said second one of said plurality of priorities.

10. (Previously Presented): The product according to claim 9, further comprising:  
computer executable instruction means for associating one of a plurality of types of requests with each one of said plurality of different priorities;

computer executable instruction means for identifying a type of each one of said plurality of HTTP requests; and

for each one of said plurality of HTTP requests, computer executable instruction means for determining one of said plurality of different priorities associated with a type that was determined for each of said plurality of HTTP requests.

11. (Canceled)

12. (Canceled)

13. (Previously Presented): The product according to claim 9, further comprising:  
computer executable instruction means for receiving said plurality of HTTP requests by said application; and

computer executable instruction means for determining one of said plurality of different priorities that is associated with a type of each one of said plurality of HTTP requests.

14. (Previously Presented): The product according to claim 9, further comprising:  
computer executable instruction means for receiving one of said plurality of HTTP requests by said application;

computer executable instruction means for determining whether there is a backlog of pending HTTP requests waiting to be processed by said application;

computer executable instruction means responsive to a determination that there is no backlog, for immediately processing said one of said plurality of HTTP requests;

computer executable instruction means responsive to a determination that there is a backlog, for determining a type of said one of said plurality of HTTP requests;

computer executable instruction means for identifying one of said plurality of priorities that is associated with said type;

computer executable instruction means for identifying one of said plurality of queues that is associated with said identified one of said plurality of priorities; and

computer executable instruction means for storing said one of said plurality of HTTP requests in said identified one of said plurality of queues.

15. (Canceled)

16. (Previously Presented): The product according to claim 9, further comprising:

computer executable instruction means for storing multiple ones of said plurality of requests having a type associated with a high priority in one of said plurality of queues that is associated with said high priority;

computer executable instruction means for storing multiple ones of said plurality of requests having a type associated with a low priority in one of said plurality of queues that is associated with said low priority; and

computer executable instruction means for completing processing of said multiple ones of said plurality of requests stored in said one of said plurality of queues that is associated with said high priority before beginning processing of said multiple ones of said plurality of requests stored in said one of said plurality of queues that is associated with a low priority.

17. (Previously Presented): A computer system executing a Web-based application, comprising:

one of a plurality of priorities being associated with each one of a plurality of different HTTP requests that are processed by an application;

a plurality of different, separate queues;

each one of said plurality of different queues being associated with a different one of said plurality of priorities;

for each one of said plurality of HTTP requests, one of said plurality of HTTP requests being stored in one of said plurality of different queues that is associated with one of said plurality of priorities that is associated with said one of said plurality of HTTP requests, wherein all of said plurality of HTTP requests that are associated with a first one of said plurality of priorities are stored in a first one of said plurality of different queues that is associated with said first one of said plurality of priorities, and all of said plurality of HTTP requests that are associated with a second one of said plurality of priorities are stored in a second one of said plurality of different queues that is associated with said second one of said plurality of priorities;

said system including a CPU executing code for completing processing of multiple ones of said plurality of HTTP requests that are stored in said first one of said plurality of different queues before beginning processing of multiple ones of said plurality of HTTP requests that are stored in said second one of said plurality of different queues regardless of whether or not said first one of said plurality of priorities is a higher priority than said second one of said plurality of priorities.

18. (Previously Presented): The system according to claim 17, further comprising:

one of a plurality of types of requests being associated with each one of said plurality of different priorities;

a type of each one of said plurality of HTTP requests being identified; and

said CPU executing code for determining, for each one of said plurality of HTTP requests, one of said plurality of different priorities associated with a type that was determined for each of said plurality of HTTP requests.

19. (Canceled)

20. (Canceled)
21. (Previously Presented): The system according to claim 17, further comprising:  
said plurality of HTTP requests being received by said application; and  
one of said plurality of different priorities that is associated with a type of each  
one of said plurality of HTTP requests being determined.
22. (Previously Presented): The system according to claim 17, further comprising:  
one of said plurality of HTTP requests being received by said application;  
said CPU executing code for determining whether there is a backlog of pending  
HTTP requests waiting to be processed by said application;  
in response to a determination that there is no backlog, said one of said plurality  
of HTTP requests being immediately processed;  
in response to a determination that there is a backlog, a type of said one of said  
plurality of HTTP requests being determined;  
one of a plurality of priorities that is associated with said type being identified;  
one of a plurality of queues that is associated with identified one of said plurality  
of priorities being identified; and  
said one of said plurality of HTTP requests being stored in said identified one of  
said plurality of queues.
23. (Canceled)
24. (Previously Presented): The system according to claim 17, further comprising:  
multiple ones of said plurality of requests having a type associated with a high  
priority being stored in one of said plurality of queues that is associated with said high  
priority;  
multiple ones of said plurality of requests having a type associated with a low  
priority being stored in one of said plurality of queues that is associated with said low  
priority; and

said multiple ones of said plurality of requests stored in said one of said plurality of queues that is associated with said high priority completing processed before processing of said multiple ones of said plurality of requests stored in said one of said plurality of queues that is associated with a low priority is begun.